

Ascent and Descent Guidance of Multistage Rockets via Pseudospectral Methods

José V. Garrido *

Instituto Superior Técnico, University of Lisbon

Marco Sagliano[†]

DLR / JAXA, Germany / Japan

This paper illustrates the trajectory modeling of multistage rockets for both ascent and descent phases. The computation of solutions is performed by coupling a multiphase optimal-control problem formulation with a transcription performed via Radau pseudospectral methods. Four examples inspired by both historical rockets like the Saturn V, and modern reusable launch systems like the Falcon 9 demonstrate the feasibility of the proposed modeling approach for the rapid prototyping of valid reference solutions.

Nomenclature

\mathcal{J}	=	Cost Functional
Φ	=	Mayer Term of the Cost Functional
Ψ	=	Lagrange Term of the Cost Functional
t	=	Independent variable
x	=	Generic state vector
u	=	Generic control vector
f	=	Derivative-free term of the dynamic equations
g	=	Algebraic nonlinear path constraints function
ϕ	=	Algebraic nonlinear endpoint constraints function
Δx	=	Phase linkage conditions
ECI	=	Earth-Centered Inertial (reference frame)
NED	=	North-East-Down

I. Introduction

Engineering applications have increased in number and in complexity throughout the years. The challenging space scenarios that engineers and scientists are facing, together with the refinement of numerical methods make the subject of optimal control of particular interest for a wide variety of modern applications. One of them is represented by Optimal control, which is of great interest as it allows a complex and detailed modeling of the problem of interest without the need of excessive simplifications which require specific assumptions. In this context the numerical advances of optimal control come at hand to help overcome these limitations.

The classical approach to solve optimal control problems is through calculus of variations, leading to the Hamiltonian Boundary-Value Problem whence numeric methods, such as shooting methods, immediately follow when analytical solutions cannot be derived. But indirect methods such as single shooting or multiple shooting are often impractical due to the associated necessity of deriving the problem-related optimality conditions. On top of this, indirect methods most often comprise convergence issues [1], and thus, lack reliability when a good initial guess is not available. For instance, indirect methods cannot deal with inequality constraints very well: typically, with shooting methods, the time intervals corresponding to the active and the inactive inequality constraints must be known a priori so that the problem can be manually divided into the respective phases. This makes indirect methods more difficult to apply in general.

*M.Sc. Graduate, Av. Rovisco Pais, 1, 1049-001, Lisboa

[†]GNC Research Engineer / Visiting GNC Specialist, 7-44-1, Chofu, Japan, AIAA Senior Member

In contrast, direct methods in general and pseudospectral methods in particular have gained popularity over the past decades since their introduction to optimal control by Elnagar *et al.* [2] and their generalization by Ross, Fahroo *et al.* [3–6]. Since then there have been multiple iterations and innovations on these methods [1, 7–10], including the implementation into several software tools such as GPOPS and GPOPS-II [11, 12], DIDO [3], PSOPT [13], and SPARTAN [14] among others. One remarkable example of pseudospectral methods application to real-world problems was the famous *zero-propellant manoeuvre* of the ISS where a large angle reorientation of the spacecraft was performed by aid of reaction wheels and gravity gradient only, with virtually no propellant spent [15]. Direct methods such as pseudospectral methods are widely used not only because they are easier to implement than indirect methods, but also because in general they can handle more complicated problems.

In a pseudospectral method, the continuous problem is discretized along the independent variable domain according to the roots of a particular Legendre polynomial; the constraints are evaluated at each discretization point resulting in a large list of equality and inequality algebraic constraints (accompanied with the algebraic objective function); this list of constraints, referred to as a nonlinear programming (NLP) problem, is passed through an off-the-shelf nonlinear optimizer such as SNOPT [16] or IPOPT [17, 18] and the solution is obtained. With the solution at hand, the covector mapping theorem (CMT) [5, 9, 19, 20] is employed to approximate the costates of the continuous problem with the multipliers associated with the NLP solution. This process allows to inspect the variational Hamiltonian and therefore the optimality of the solution. Finally, having the discretized solution, the continuous interpolation can be obtained by use of the Lagrange interpolating polynomials in post-processing. Pseudospectral methods are known to converge spectrally [5], which implies that for most problems the discrete solution converges very quickly to the continuous one with the increase of the number of nodes used to solved the problem.

It is well known that direct methods, including pseudospectral methods deeply benefit from an efficient computation of the Jacobian matrix, and in some cases even from the Hessian matrix. This information can be provided by the user in analytic form, but this process is error-prone and determines a significant burden, especially for larger and complicated problems. A more efficient way is to provide the differential information via systematic numerical method. This approach implies that software packages include numerical differentiation algorithms to find the partial derivatives of the constraints with respect to the variables and ultimately assemble the numeric Jacobian matrix. In the context of optimal control, approaches such as the *complex step* method and the *Dual number* method have been successfully employed over the last years [21–24].

Another relevant aspect is the numerical conditioning of the problem. Also in this case it is possible to choose between a problem-defined ad-hoc scaling and a numerically-driven process to systematically scale the constraints and the Jacobian matrix of the problem. One effective scaling method is to factor each constraint by the magnitude of the respective Jacobian matrix row [23, 25, 26]. Despite not introducing coupling between the constraints, the scaling does affect the resulting output dual variables. This means that the NLP needs to be rescaled back in post-processing before the final solution output and also before the covector mapping is applied [23].

Within pseudospectral methods, the three most commonly used techniques are the Legendre-Gauss pseudospectral method, the Legendre-Gauss-Lobatto pseudospectral method and the Legendre-Gauss-Radau pseudospectral method (in its direct or flipped version), all of which have been properly formalized [9]. The roots of Legendre polynomials sit invariably within the interval $[-1, +1]$. The position of the roots changes slightly from one method to another, but the most relevant difference between the methods is the inclusion (or exclusion) of the interval endpoints (-1) or (+1). The roots of the Legendre-Gauss polynomial do not include any of the endpoints; the roots of the Legendre-Gauss-Lobatto polynomial include both of them; and the roots of the Legendre-Gauss-Radau polynomial include either the left endpoint (-1) (direct Radau) or the right endpoint (+1) (flipped Radau) of the interval.

On the one hand, because the Legendre-Gauss polynomial does not include the endpoints as roots, the respective method by itself does not allow for corresponding boundary nonlinear constraints, and in order to include this type of conditions, one must add an auxiliary quadrature constraint for artificial collocation of the terminal (or initial) endpoint [11]. This extra constraint constitutes an additional weight on the NLP, making the problem slightly, but unnecessarily, more difficult to solve. On the other hand, the Legendre-Gauss-Lobatto method does contain both endpoints of the interval and so it does provide the possibility to include boundary nonlinear constraints by itself; however it has been shown that this method has convergence issues with respect to the dual variables [10], which compromises the eventual computation of the Hamiltonian.

Regarding multiple phase optimal control, successful efforts have been made to implement the feature into software [11, 12]. Multiphase optimal control allows the study of trajectories that might contain necessary state discontinuities or even known mutations of the equations of motion. This feature are particularly interesting from the point of view of multiple stage Orbital Launch Vehicles (OLVs).

Other developments in pseudospectral optimal control have taken place, most notably h and hp mesh refinement methods [24, 27–30]. In a h method the number of polynomials to be concatenated along the domain is increased, while in a p method the degree of the polynomial in use is increased. hp methods are a combination of the two. The implementation of p mesh refinement methods is often superfluous, as it is often more satisfying to the user to manually control the degree of the polynomial at rerun whenever a solution has not converged to the desired accuracy. On the other hand methods h and hp can be powerful, however, the respective implementation can be laborious and, for most cases, the refinement is unnecessary.

In this paper the flipped Radau pseudospectral method is employed to solve multiple phase optimal control problems, more specifically, the method is used to optimize trajectories of multiple stage OLVs in both ascension to orbit and descent to vertical landing. The Legendre-Gauss-Radau method is well suited because it is a pseudospectral method that allows endpoint collocation, and it does not present dramatic convergence issues on the dual variables. The foundation for the implementation of the multiphase feature is SPARTAN [14], a MATLAB tool developed by DLR (Deutsches Zentrum für Luft- und Raumfahrt). The algorithm is shown to be reliable for both types of multiple phase problem, namely ascent and descent guidance problems, as the ones presented here as numerical examples.

The present paper is organized as follows. In Sec. II the problem formulation is addressed: the optimal control problem is exposed and the generic equations of motion expressed in such way to account for both types of trajectory: lift-off and ascension to orbit; and descent to vertical landing. Section III is dedicated to the exposition of the multiphase algorithm, namely the discretization process as well as the resulting structure of the Jacobian matrix. Numerical examples are presented in Sec. IV as demonstration of the program. Finally, concluding remarks are drawn in Sec. V.

II. Ascent and Descent Guidance Problem

In this section a brief description of ascent and descent problems is presented in a generic form. Firstly, the formulation of a multiple phase optimal control problem is expressed as to set the context for trajectory optimization. After that the equations of motion for rockets and rocket-like dynamic systems are presented in generic form in order to be adequate to apply to both ascent- and descent-trajectory types.

A. The multiple phase optimal control problem

A trajectory can only be classified as optimal with respect to a relevant associated performance measure. The performance measure is a scalar quantity that serves as the indicator of optimality. In optimal control, this quantity is conventionally expressed as a cost functional, \mathcal{J} , this way the optimal trajectory is found by minimizing the cost. An optimal control problem exists in the domain of an independent variable about which integration occurs. In the context of dynamic systems such as orbital launch vehicles the independent variable, t , is set to be the continuous time. In this way, letting $p \in [1, \dots, P]$ be a phase indicator, where P is the total number of phases, a generic multiple phase optimal control problem can be formulated as follows.

Minimize the cost functional

$$\mathcal{J} = \Phi(\mathbf{x}(t_0^{(1)}), t_0^{(1)}, \mathbf{x}(t_f^{(P)}), t_f^{(P)}) + \sum_{p=1}^P \left\{ \int_{t_0^{(p)}}^{t_f^{(p)}} \Psi(\mathbf{x}(t), \mathbf{u}(t), t) dt \right\} \quad (1)$$

associated with the trajectory of the multiphase dynamic system

$$\dot{\mathbf{x}}(t) = \mathbf{f}^{(p)}(\mathbf{x}(t), \mathbf{u}(t), t), \quad p = 1, \dots, P, \quad (2)$$

subject to the path constraints

$$\mathbf{g}_{min}^{(p)} \leq \mathbf{g}^{(p)}(\mathbf{x}(t), \mathbf{u}(t), t) \leq \mathbf{g}_{max}^{(p)}, \quad p = 1, \dots, P, \quad (3)$$

the event constraints

$$\phi_{min}^{(p)} \leq \phi^{(p)}(\mathbf{x}(t_0^{(p)}), t_0^{(p)}, \mathbf{x}(t_f^{(p)}), t_f^{(p)}) \leq \phi_{max}^{(p)}, \quad p = 1, \dots, P, \quad (4)$$

and the phase linkage conditions

$$\Delta \mathbf{x}_{min}^{(p)} \leq \Delta \mathbf{x}^{(p)} = \mathbf{x}(t_0^{(p+1)}) - \mathbf{x}(t_f^{(p)}) \leq \Delta \mathbf{x}_{max}^{(p)}, \quad p = 1, \dots, P-1, \quad (5)$$

The phases are assumed to be sequential and the interface times from one phase to the next are assumed to be identical $t_0^{(p+1)} = t_f^{(p)}$, $p = 1, \dots, P - 1$. This formulation, although seemingly not completely generic, is actually sufficient to account for a vast amount of optimal control problems.

For context, a list of scenarios that might constitute justification for phase transitions is presented:

- i State discontinuities at interior points
- ii Mutations of the equations of motion
- iii Mutation of the algebraic path constraints
- iv Nonlinear event constraints at interior points

It is relevant to notice that, with the exception of the phase-linkage conditions, the constraints of a multiple phase optimal control problem are always phase-wise decoupled.

B. The cost functional for spacecraft trajectories

Recalling the cost functional from equation (1), there are a few notable remarks that we address now.

When $\Phi \equiv 0$ the problem is known as a *Lagrange problem*, and when $\Psi \equiv 0$ it is known as a *Mayer problem*. It is in general possible to express a Lagrange problem as a Mayer problem (and vice-versa). Despite this, the Bolza is the standard formulation of a generic optimal control problem, standing as the sum of an integral functional of the continuous problem and a function at the boundaries, and includes Mayer and Lagrange forms as special cases.

The performance measure is the parameter that dictates the optimality, and reflects some physical quantities to be minimized or maximized to cope with the requirements of a real system. For example, depending on the scenario and the mission constraints we might be interested to minimize final time or final energy of a spacecraft. With the two trajectories different from each other, yet both being optimal. Therefore, the process of finding the optimal trajectory begins by choosing the optimal performance measure that best serves the physical nature of the problem we are dealing with. In the case of OLV a few scenarios are common:

- i Maximization of the final mass: $\mathcal{J} = -m(t_f)$;
- ii Minimization of the time duration of the trajectory: $\mathcal{J} = t_f - t_0$, $t_f > t_0$;
- iii Maximization of displacement: $\mathcal{J} = -r(t_f)$;
- iv Minimization of the error about the objective displacement: $\mathcal{J} = (r(t_f) - r_{obj})^2$.

These scenarios are representative of all the cases addressed in this paper.

C. Equations of motion for rocket-like dynamic systems

The propulsive system of a rocket is based on the reactive force from mass exhaust. The general dynamic equations for systems subject to mass variations of this kind can be expressed as [31]:

$$\mathbf{F}_{net} = \mathbf{F}_{ext} + \mathbf{T} = m \frac{d\mathbf{v}}{dt} \quad (6)$$

$$\mathbf{T} = \mathbf{v}_e \frac{dm}{dt} \quad (7)$$

where the vector \mathbf{F}_{net} is the resulting net force acting on the vehicle, the vector \mathbf{F}_{ext} is the sum of common external forces that do not change the mass of the system, \mathbf{T} is the thrust vector, m is the total mass of the vehicle, \mathbf{v} is the velocity vector, and \mathbf{v}_e is the effective mass exhaust velocity vector. Eqs. (6) and (7) account for propulsion based on escaping mass, $\frac{dm}{dt} < 0$, and on incident mass, $\frac{dm}{dt} > 0$, with respect to the center of mass of the vehicle. Given that $\frac{dm}{dt}$ is a scalar quantity, it is evident that the thrust vector, \mathbf{T} , is always collinear with the exhaust velocity vector, \mathbf{v}_e .

The equations of motion for rockets and other rocket-like propulsive systems can be written without loss of generality by solving Eqs. (6) and (7) for the rates of change of velocity and mass, respectively, $\frac{d\mathbf{v}}{dt}$ and $\frac{dm}{dt}$, and adjoining the kinematic equations ($\dot{\mathbf{r}} = \mathbf{v}$), where \mathbf{r} is the position vector:

$$\dot{\mathbf{r}} = \mathbf{v} \quad (8)$$

$$\dot{\mathbf{v}} = \frac{\mathbf{F}_{ext}(\mathbf{r}, \mathbf{v})}{m} + \frac{\mathbf{T}}{m} \quad (9)$$

$$\dot{m} = -\frac{\|\mathbf{T}\|}{v_e} \quad (10)$$

The generic force $\mathbf{F}_{ext}(\cdot)$ is representative of external sources of acceleration such as gravitational forces, aerodynamic forces, or fictitious forces introduced by a non-inertial reference frame. The norm of the effective exhaust velocity vector, $v_e = \|\mathbf{v}_e\|$, is a known constant. In the particular case of constant $\|\mathbf{T}(t)\|$ the mass of the vehicle will decay linearly with time.

Most often the thrust vector represents the control of the system, $\mathbf{T}(t) = \mathbf{u}(t)$, but in some cases the thrust can be defined as a state instead. This mutation is necessary for a better modelling of the behaviour of the engine and it implies the presence of the additional equation $\dot{\mathbf{T}}(t) = \mathbf{u}(t)$ to Eqs. (8)-(10).

Equations (8) to (10) presume a point-mass approximation of the system, this is to say that these equations overlook some relevant aspects of rocket behaviour such as attitude control. More refined approximations imply the extension of the state vector and the consequent augmentation of the equations of motion.

D. Path constraints

In addition to the difficulties solely imposed by the dynamic nature of multi-stage rockets, an OLV might be subject to certain quantitative conditions in flight due to design requirements or safety factors. For instance, it might be necessary to keep the aerodynamic pressure below a certain value along the trajectory, or it might be the case that the acceleration experienced by the vehicle must be limited to a certain level, or, still, the trajectory might be physically conditioned due to nearby no-fly zones.

These types of constraints are called *path constraints* and they can be thought of as an algebraic equation (or inequality) that must be verified at each point in time.

For reference, other examples of path constraints commonly used for convenience reasons are:

- To assert the norm of the position vector to be greater than or equal to the radius of the Earth, in order to avoid collision courses.
- Guarantee Pythagorean relationship between Cartesian components of thrust, in order to assert constant vector norm.
- Include geometric constraints between spatial coordinates in order to avoid optimal solutions that do not satisfy further requirements (e.g., safe or environmental constraints).

III. Multi-phase flipped Radau Pseudospectral Method

In this section the transcription according to the flipped Radau method is explained in detail. The transcription process typically involves the discretization of the continuous domain of the problem, a differential operator to approximate the equations of motion, as well as an integral operator used to properly represent the Lagrange term of the cost function. In the following subsections these steps will be described.

A. Transcription

The discretization of the optimal control problem is necessary so that a computer program can be tasked to solve the associated NLP problem. The discretization is done by sampling the variables and the functions at multiple points along the independent variable.

Given a phase discretized at $N + 1$ points along the independent variable, a convenient assumption to make is that the continuous solution will be well approximated by an explicit polynomial function of degree N . Then, the smooth polynomial function of degree N which passes through the $N + 1$ points can be found via *Lagrange polynomial interpolation*,

$$\mathbf{x}(t) \approx \mathbf{X}(t) = \sum_{i=0}^N \mathbf{X}(t_i) L_i(t) \quad L_i(t) = \prod_{\substack{j=0 \\ j \neq i}}^N \frac{t - t_j}{t_i - t_j}$$

where $\mathbf{x}(t)$ is the real solution, $\mathbf{X}(t)$ is the continuous polynomial approximation, t_i is the independent variable associated with the i^{th} sample point and L_i is the Lagrange interpolating polynomial associated with the i^{th} sample point.

Furthermore, if the solution is well approximated by the N degree polynomial $\mathbf{X}(t)$, the first derivative of the solution, with respect to the independent variable can also be estimated based on the properties of Lagrange polynomials, as in [32]:

$$\frac{d\mathbf{x}(t)}{dt} \approx \frac{d\mathbf{X}(t)}{dt} = \sum_{i=0}^N \mathbf{X}(t_i) \frac{dL_i(t)}{dt} \quad (11)$$

In discrete terms, Eq. (11) is equivalent to the matrix-vector multiplication between a *differentiation matrix* and the column vector of the sampled points X_i ,

$$\dot{X}_k = \sum_i D_{ki} X_i, \quad i = 0, 1, 2, \dots, N, \quad k \subseteq i \quad (12)$$

where the elements of the differentiation matrix are defined as:

$$D_{ki} = \dot{L}_i(t_k) = \sum_{\substack{l=0 \\ l \neq i}}^N \frac{1}{t_i - t_l} \prod_{\substack{j=0 \\ j \neq i, l}}^N \frac{t_k - t_j}{t_i - t_j} \quad (13)$$

The operation expressed in Eq. (12) computes the derivative of X at the points t_k as linear combination of the state X evaluated at the points t_i .

Similarly, assuming once again that the solution is well approximated by a polynomial function, it is possible to compute a definite integral over a specified interval of the independent variable according to a Gaussian quadrature, simply by evaluating the argument function at the discrete points and computing a weighted sum, as in [9]:

$$\int_{-1}^1 f(t) dt = \sum_{k=1}^N w_k f(t_k) \quad (14)$$

The three discrete operations (the Lagrange polynomial interpolation, the derivative operator by differentiation matrix multiplication, and the integration by Gaussian quadrature) above described can be more or less accurate according to a wiser or poorer choice of sample points t_k along the domain of the independent variable, respectively.

B. Flipped Radau polynomial roots

Due to the Runge phenomenon, the worst possible choice of sample points for polynomial interpolation is an equidistant grid [9, 23]. In contrast, a good choice of sample points is a proportional mapping of the roots of Legendre-based polynomials, such as the *flipped Radau polynomial*. The Flipped Radau Polynomial is the result of the difference between two Legendre polynomials of consecutive degree:

$$R_N(\tau) = P_N(\tau) - P_{N-1}(\tau), \quad \tau \in [-1, 1] \quad (15)$$

where $P_N(\tau)$ is the Legendre polynomial of order N .

Figure 1 shows the comparison between an equidistant grid and the distribution of the flipped Radau polynomial roots for reference. One peculiarity of the roots of the Radau polynomial is that they are not symmetric with respect to the origin.

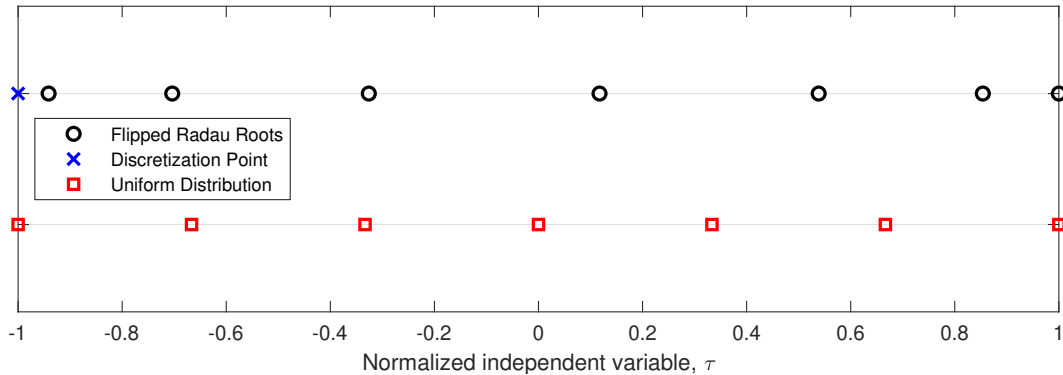


Fig. 1 Comparison of node distributions for $N = 7$ in the domain $[-1, 1]$. A uniform distribution is shown in red squares. The roots of the flipped Radau polynomial are shown in black circles, an extra discretization point at $\tau = -1$ is shown as a blue cross.

The Gaussian quadrature weights associated with the roots of the flipped Radau polynomial can be computed by doing a *flip* operation on the weights of the direct Radau, as in [9, 23]:

$$w_k = \text{flip} \left\{ \frac{1 - \tau_k}{N^2 P_{N-1}^2(\tau_k)} \right\}, \quad k = 1, 2, \dots, N \quad (16)$$

where τ_i is the i^{th} abscissa of the direct Radau roots; N is the degree of the Radau polynomial; and $P_{N-1}(\tau_i)$ is the Legendre polynomial of degree $N - 1$ evaluated at τ_i . The Gaussian quadrature associated with the flipped Radau method is accurate for polynomial functions of degree up to $2N - 2$.

Because the roots of the flipped Radau polynomial do not include the left limit of the interval ($\tau = -1$) there will be no weight associated with this point, and therefore this point cannot be collocated. However the point at $\tau = -1$ can be used as an auxiliary discretization sample for the calculus of the derivative at the collocation points. Also, this additional discretization point allows the specification of the state at the beginning of the trajectory (initial condition).

Letting $i = 0, 1, 2, \dots, N$ be the indices corresponding to the discretization points, and $k = 1, 2, \dots, N$ be the indices corresponding to the collocated points, the resulting matrix \mathbf{D} from Eq. (13) will be rectangular of size $N \times (N + 1)$.

Moreover, by mapping the domain of the independent variable, $t \in [t_0, t_f]$, into the normalized domain of $\tau \in [-1, 1]$, as follows [23].

$$t = \frac{t_f - t_0}{2} \tau + \frac{t_f + t_0}{2} \quad \tau = \frac{2}{t_f - t_0} t + \frac{t_f + t_0}{t_f - t_0}, \quad (17)$$

The integral operation from Eq. (14) can be generalized for an arbitrary interval of integration:

$$\int_{t_0}^{t_f} f(t) dt = \frac{t_f - t_0}{2} \int_{-1}^1 f(t(\tau)) d\tau = \frac{t_f - t_0}{2} \sum_{k=1}^N w_k f(t_k) \quad (18)$$

where $t_k = t(\tau_k)$ and $t_N = t_f$.

The sampling illustrated in Fig. 1 for the Radau roots is applied to each phase individually because each phase is smooth and differentiable, and because the phases are decoupled from one another (derivatives and integral operators act only about the respective phase).

C. Constraints in discrete algebraic form

By taking advantage of the discrete operators discussed in section III.A and the properties associated with the flipped Radau discretization from section III.B, a list of algebraic constraints can be formulated that will be an equivalent expression of the initial continuous optimal control problem from section II.A, and which can be used as input to the nonlinear solver.

The discrete cost functional is expressed as

$$\mathcal{J}^N = \Phi[X_0^{(1)}, t_0^{(1)}, X_N^{(P)}, t_N^{(P)}] + \sum_{p=1}^P \frac{t_N^{(p)} - t_0^{(p)}}{2} \sum_{k=1}^{N^{(p)}} w_k^{(p)} \Psi[X_k^{(p)}, U_k^{(p)}, t_k^{(p)}], \quad k = 1, \dots, N^{(p)} \quad (19)$$

The integral term (Lagrange cost) turns into the sum of weighted evaluations of the argument function Ψ , as expressed in Eq. (18), and the sum across phases is performed. The number of nodes, N , can be specified for each phase individually, and that implies that the quadrature weights might also change from phase to phase.

Next, we compute the differentiation matrix associated with the nodes of the Radau polynomial, Eq. (13), and by performing the operation stated in Eq. (12), equality constraints can be formulated such that the dynamic defects are asserted to be zero, as in [5, 11, 14]:

$$\mathbf{F}^{(p)} = \frac{t_N^{(p)} - t_0^{(p)}}{2} f_k^{(p)}[X_k^{(p)}, U_k^{(p)}, t_k^{(p)}] - \sum_{i=0}^{N^{(p)}} \mathbf{D}_{ki}^{(p)} X_i^{(p)} = \mathbf{0}, \quad (20)$$

$$i = 0, 1, \dots, N^{(p)}, \quad k = 1, \dots, N^{(p)}$$

where the summation term simply represents the inner product between the discrete vector X_i and the k^{th} row of the differentiation matrix D .

Regarding the path constraints and the event constraints, these are expressed in general terms as inequality constraints, and the only requirement is that they lie within the user-specified boundaries.

$$\mathbf{g}_{min}^{(p)} \leq \mathbf{g}^{(p)} \left[\mathbf{X}_k^{(p)}, \mathbf{U}_k^{(p)}, t_k^{(p)} \right] \leq \mathbf{g}_{max}^{(p)} \quad (21)$$

$$\boldsymbol{\phi}_{min}^{(p)} \leq \boldsymbol{\phi}^{(p)} \left[\mathbf{X}_0^{(p)}, t_0^{(p)}, \mathbf{X}_N^{(p)}, t_N^{(p)} \right] \leq \boldsymbol{\phi}_{max}^{(p)} \quad (22)$$

The linkage conditions are the only set of constraints which introduces algebraic coupling of variables between different phases, in practice, these constraints only represent additional boundary conditions that must be satisfied and which relate the states of one phase to the next. Linkage conditions are therefore, a source of discontinuities in the problem, and they do not influence the continuous optimality conditions. The linkage conditions in discrete algebraic form are expressed as:

$$\Delta \mathbf{x}_{min}^{(p)} \leq \Delta \mathbf{X}^{(p)} = \mathbf{X}_0^{(p+1)} - \mathbf{X}_N^{(p)} \leq \Delta \mathbf{x}_{max}^{(p)} \quad (23)$$

D. Adjoined vector of decision variables, vector of constraints and Jacobian matrix

In order to construct a nonlinear programming problem one needs both, a list of decision variables and a list of algebraic constraints. Let

$$\mathbf{XU} = \left[\mathbf{X}_0^T \quad \mathbf{X}_1^T \quad \mathbf{U}_1^T \quad \mathbf{X}_2^T \quad \mathbf{U}_2^T \quad \dots \quad \mathbf{X}_N^T \quad \mathbf{U}_N^T \right] \quad (24)$$

be the concatenated state-control decision vector associated with every node of a given phase p . Notice that there is no control vector associated with the very first node $i = 0$, again this is because this node is not collocated. The resulting size of this row vector is $1 \times [n_x + N^{(p)}(n_x + n_u)]$.

The complete vector of decision variables can be expressed as:

$$\mathbf{X}_{NLP} = \left[\mathbf{XU}^{(1)} \quad \mathbf{XU}^{(2)} \quad \dots \quad \mathbf{XU}^{(P)} \quad t_N^{(1)} \quad t_N^{(2)} \quad \dots \quad t_N^{(P)} \right]^T \quad (25)$$

resulting in a column vector of size $[P(n_x + 1) + (n_x + n_u) \sum_{p=1}^P N^{(p)}] \times 1$.

Further, taking the algebraic constraints expressed in section III.C, we may express the list of constraints as follows:

$$\mathbf{C}_{NLP} = \left[\mathcal{J}^N \quad \mathbf{F}^{(1)} \quad \dots \quad \mathbf{F}^{(P)} \quad \mathbf{g}^{(1)} \quad \dots \quad \mathbf{g}^{(P)} \quad \boldsymbol{\phi}^{(1)} \quad \dots \quad \boldsymbol{\phi}^{(P)} \quad \Delta \mathbf{X}^{(1)} \quad \dots \quad \Delta \mathbf{X}^{(P-1)} \right]^T \quad (26)$$

The vectors \mathbf{X}_{NLP} and \mathbf{C}_{NLP} describe the nonlinear programming problem at hand. Thus, it is possible to build a Jacobian matrix that will aid the nonlinear solver into finding a solution.

The Jacobian matrix is a quantitative description of the derivatives of the constraints with respect to the decision variables, providing a first order gradient that is valuable for the non linear solver.

Each row of the Jacobian matrix corresponds to an algebraic constraint, and each column corresponds to a decision variable. Ultimately, each element represents a linear dependency a constraint (row) with respect to a decision variable (column). Mathematically the Jacobian is expressed as:

$$\mathbf{Jac} = \nabla_{\mathbf{X}_{NLP}} \mathbf{C}_{NLP} = \nabla \mathbf{C}_{NLP} = \begin{bmatrix} \nabla \mathcal{J}^N \\ \nabla \mathbf{F}^{(1:P)} \\ \nabla \mathbf{g}^{(1:P)} \\ \nabla \boldsymbol{\phi}^{(1:P)} \\ \nabla \Delta \mathbf{X}^{(1:P-1)} \end{bmatrix} \quad (27)$$

The format of the \mathbf{X}_{NLP} vector from Eq. (25) dictates the sparsity pattern of the Jacobian matrix. Figure 2 illustrates the pattern of the Jacobian for an example problem with four phases: the first row of the matrix corresponds to the discrete cost functional where the magenta circles represent a Lagrangian cost and the blue dot at the end of phase four represents a Mayer cost; there are four open terminal times (green columns); a scalar path constraint applied to all phases (concatenated diagonals in cyan); vector event constraints at the terminal times of every phase (magenta

blocks); and three linkage conditions connecting the four phases sequentially (red diagonals at the bottom rows). The four phases can be distinguished by the four large red blocks with blue diagonal smaller blocks (corresponding to the dynamic defects of each phase).

It is relevant to note that the diagonals on the linkage conditions rows are, invariably, positive and negative identity matrices with the dimension of the state vector, $\pm I_{n_x}$, connecting one phase to the next. Also, it is noticeable in Fig. 2, that the constraints of different phases are decoupled from one another, in other words, decision variables of one phase do not affect the constraints of any other phase, thus the Jacobian matrix is naturally very sparse.

IV. Numerical examples

This section shows some numerical examples for the described methodology. We propose four different study cases: for the ascent we considered a multi-stage solid-propellant engine fictional rocket taken from literature, which served as prototyping example for the methodology, and the ascent of the historical Saturn-V Rocket based on the documentation available for the Apollo 11 mission. For the descent we consider again a fictional example taken from literature and a more realistic example inspired by SpaceX's Falcon 9 mission profile for the recovery of the first stage.

A. Solid-propellant Engine Multistage Rocket

This example deals with the optimization of the ascent trajectory of a solid propellant multi-staged rocket taken from Refs. [11, 12, 33]. This problem is appropriate to validate the algorithm developed in this work because it is composed of four distinct phases ($P = 4$). The modelling of the problem in four phases results from mutations in the equations of motion from one phase to the next which yield removable discontinuities on the velocity, but also, and most remarkably, from jump discontinuities in the mass of the vehicle due to stage separation events.

This problem is about finding the trajectory from launch pad to orbit insertion of a vehicle composed of 9 solid rocket boosters, 1 main stage, and 1 upper stage. The flight sequence is as follows:

- Phase 1 Ignition of the main stage and of 6 solid boosters (out of 9) at $t = t_0$. Full throttle until the burnout of the 6 solid boosters. The 6 empty boosters are separated and dropped out at $t = t_1$.
- Phase 2 Main stage continues in full throttle during all of Phase 2. Ignition of the 3 left over boosters at $t = t_1$. Full throttle until the burnout of the boosters. 3 empty boosters are separated and dropped out at $t = t_2$.
- Phase 3 Main stage continues in full throttle. Burn out of the remaining fuel in the main stage. Main stage is separated and dropped out at $t = t_3$.
- Phase 4 Ignition of the upper stage at $t = t_3$. Full throttle until orbit insertion. Payload deployed at $t = t_4$.

The equations of motion are taken directly from the original article and are expressed in ECI coordinates. The Earth is assumed to be spherical and rotating, and the acceleration of gravity decays with the radial distance to the center of the planet by the inverse square law.

The optimal control problem is stated as follows: minimize the cost functional

$$\mathcal{J} = -m(t_f) \quad (28)$$

subject to the equations of motion

$$\dot{\mathbf{r}} = \mathbf{v} \quad (29)$$

$$\dot{\mathbf{v}} = -\frac{\mu}{\|\mathbf{r}\|^3}\mathbf{r} + \frac{T}{m}\mathbf{u} + \frac{\mathbf{D}}{m} \quad (30)$$

$$\dot{m} = -\frac{T}{g_0 I_{sp}} \quad (31)$$

the initial conditions

$$\mathbf{r}(t_0^{(1)}) = [5605.2 \times 10^3 \quad 0 \quad 3043.4 \times 10^3]^\top \text{m} \quad (32)$$

$$\mathbf{v}(t_0^{(1)}) = [0 \quad 0.4076 \times 10^3 \quad 0]^\top \text{m/s} \quad (33)$$

$$m(t_0^{(1)}) = 301\,454 \text{ kg} \quad (34)$$

$$(35)$$

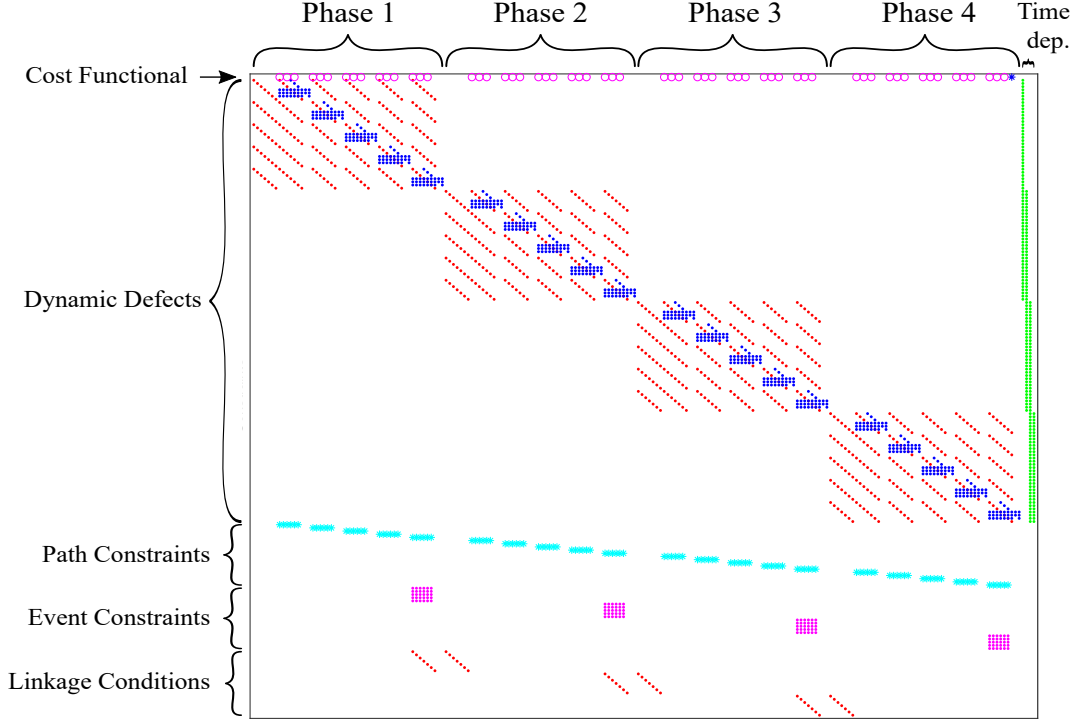


Fig. 2 Sparsity pattern of the Jacobian matrix for an example problem with 4 phases, 5 collocation points on each phase. 4 open terminal times. One scalar path constraint applying to every phase and a terminal constraint vector applying to every phase.

the terminal conditions (orbital elements)

$$a_f^{(4)} = 24\,361.14 \text{ km (semi-major axis)} \quad (36)$$

$$e_f^{(4)} = 0.7308 \text{ (eccentricity)} \quad (37)$$

$$i_f^{(4)} = 28.5 \text{ deg (inclination)} \quad (38)$$

$$\Omega_f^{(4)} = 269.8 \text{ deg (longitude of ascending node)} \quad (39)$$

$$\omega_f^{(4)} = 130.5 \text{ deg (argument of perigee)} \quad (40)$$

$$v_f^{(4)} = \text{free (true anomaly)} \quad (41)$$

and the phase linkage conditions

$$\mathbf{r}(t_0^{(p+1)}) - \mathbf{r}(t_f^{(p)}) = \mathbf{0} \quad (42)$$

$$\mathbf{v}(t_0^{(p+1)}) - \mathbf{v}(t_f^{(p)}) = \mathbf{0} \quad (43)$$

$$m(t_0^{(p+1)}) - m(t_f^{(p)}) = -m_{dry}^{(p)} \quad (44)$$

Where $\mathbf{r}, \mathbf{v}, m$, are, respectively, position, velocity and total mass of the vehicle, μ is the standard gravitational parameter of the Earth, T is the thrust modulus (constant along each phase), g_0 is the Earth's gravity acceleration, I_{sp} is the specific impulse (constant along each phase) and D is the aerodynamic drag force.

The drag force assumes an exponential model of the atmospheric air density and it is calculated as follows,

$$\mathbf{D} = -\frac{1}{2}\rho_0 e^{-\frac{h}{H}} S C_D \|\mathbf{v}_{rel}\| \mathbf{v}_{rel} \quad (45)$$

with

$$h = \|\mathbf{r}\| - R_e \quad (46)$$

$$\mathbf{v}_{rel} = \mathbf{v} - \begin{bmatrix} 0 & 0 & \Omega_{\oplus} \end{bmatrix}^T \times \mathbf{r} \quad (47)$$

and where q is the dynamic pressure, ρ is the air density as a function of altitude, ρ_0 is the standard air density at sea level, h is the altitude of the spacecraft, H is the density scale height, S is the reference surface area, C_D is the drag coefficient, \mathbf{v}_{rel} is the velocity of the spacecraft relative to the atmosphere, R_e is the radius of the earth and Ω_{\oplus} is the angular velocity of Earth's rotation. Tables 1 and 2 show the values of all the constants and vehicle properties mentioned above.

It is relevant to note that the intermediate phase transition times t_1 , t_2 and t_3 are known (fixed), but the final time of phase 4 is open and to be determined. The problem was solved with the collocation of 8, 8, 8, and 16 flipped-Radau nodes on phases 1, 2, 3 and 4, respectively. The NLP solver used was IPOPT [18], and the computation of partial derivatives was done through the complex step differentiation method [21, 34].

Table 1 Relevant constants and parameters for the multistage solid propellant rocket problem.

Constant	Value	Unit
Payload mass, $m_{payload}$	4164	kg
Reference surface area, S	4π	m ²
Drag Coefficient, C_D	0.5	–
Air density at sea level, ρ_0	1.225	kg/m ³
Density scale height, H	7200	m
Earth radius, R_e	6378145	m
Earth rotation rate, Ω_{\oplus}	$7.292\,115\,85 \times 10^{-5}$	rad/s
Standard gravitational parameter, μ	$3.986\,012 \times 10^{14}$	m ³ /s ²
Gravity acceleration at sea level, g_0	9.80665	m/s ²
Phase 1 initial time, t_0	0	s
Phase 1 / Phase 2 transition time, t_1	75.2	s
Phase 2 / Phase 3 transition time, t_2	150.4	s
Phase 3 / Phase 4 transition time, t_3	261	s

Source: 2014, Michael A. Patterson et al. [12].

Table 2 Properties of each component of the vehicle for the multistage solid propellant rocket problem.

	Solid Rocket Boosters	Main Stage	Upper Stage	Unit
Total Mass, $m_{component}$	19290	104380	19300	kg
Propellant Mass, m_{prop}	17010	95550	16820	kg
Engine Thrust, T	628500	1083100	110094	N
Burn Time, t_b	75.2	261	700	s
Number of Engines	9	1	1	–
Specific Impulse, I_{sp}	$T t_b / (g_0 m_{prop})$			s

Source: Anil V. Rao et al. (2010) [11]

The results for this problem are presented in Figs. 3 through 9.

Figures 3 and 4 show the altitude of the spacecraft and the norm of the velocity vector with time, respectively. It can be seen that the altitude, expressed with respect to the Earth radius, starts at 0 (zero) with a path that is tangential to the time axis, and goes to about 200 km where orbit insertion occurs. The profile of the altitude seems to "wobble" during the last phase where it decreases after reaching a local maximum, only to increase again for the orbit insertion point. The velocity norm starts a little above zero due to the tangential component introduced by the Earth's rotation at the surface and it follows a non-decreasing profile along the trajectory.

Figures 5 and 6 present a direct comparison between the results obtained with SPARTAN and the solution of the original article [11]. The figures shows the same variables depicted in Fig. 3 and Fig. 4, respectively: altitude norm

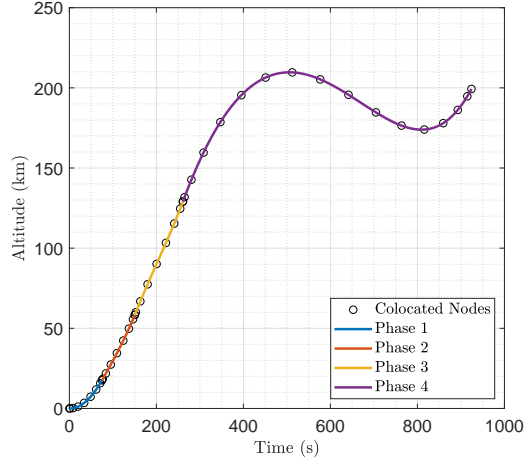


Fig. 3 Example IV.A- Altitude vs time.

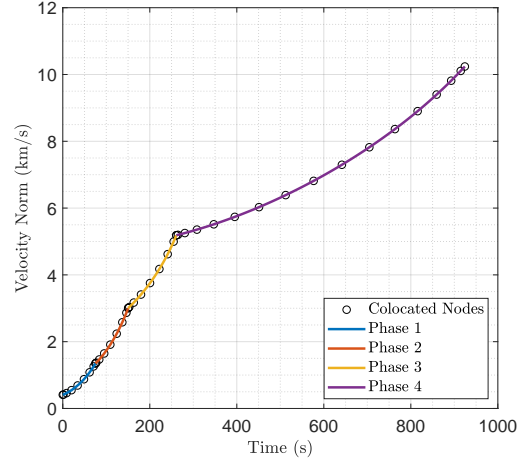


Fig. 4 Example IV.A- Velocity norm vs time.

of velocity. The two solutions are superimposed for direct comparison with the state-of-the-art solver GPOPS. With regards to both plots, it can be seen that the two solutions show excellent agreement with each other, being virtually indistinguishable.

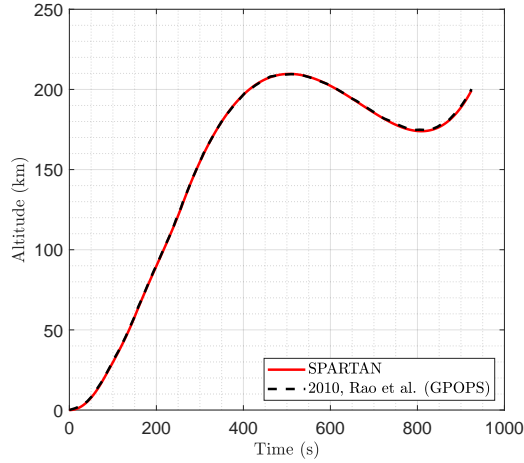


Fig. 5 Example IV.A- Altitude vs time. Comparison of results [11].

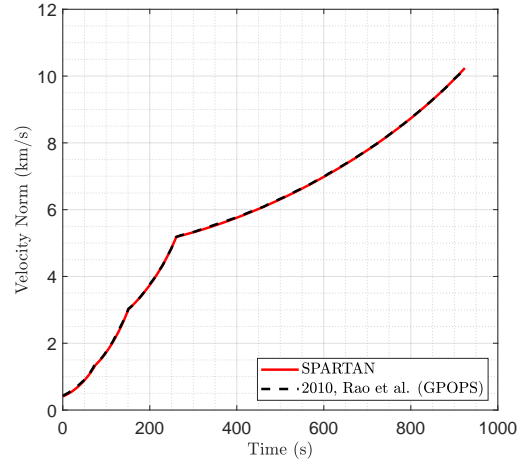


Fig. 6 Example IV.A- Velocity norm vs time. Comparison of results [11].

The profile of the total mass of the vehicle is shown in Fig. 7. In agreement with the equations of motion, the mass decays linearly due to constant mass flow rate on each phase (steeper mass gradients correspond to larger mass flow rates). Also, it is possible to see that the plot presents clear jump discontinuities at the phase transitions, this is the direct result of stage separation events where empty rocket boosters are discarded. Jump discontinuities in the mass of the system represent a main motivation for the employment of a multiphase algorithm. This characteristic of the problem makes it adequate to validate the flipped Radau pseudospectral method.

The Cartesian components of the thrust (control) are presented in Fig. 8. In this case the phase transition points are evidenced by the increased density of colocated nodes along the time domain. It can be asserted that the components assert a unit vector throughout the trajectory. The representation of thrust in unit vector form is convenient to preserve continuity, as the modulus of thrust changes instantaneously at every phase (the thrust is subject to jump discontinuities). The smoothness of the thrust unit vector along the trajectory indicates that the thrust vector, although subject to jump discontinuities in its norm, preserves smoothness in its direction regardless of the phase transitions.

It is well known that every optimal control problem can be formulated in multiple ways without changing what will

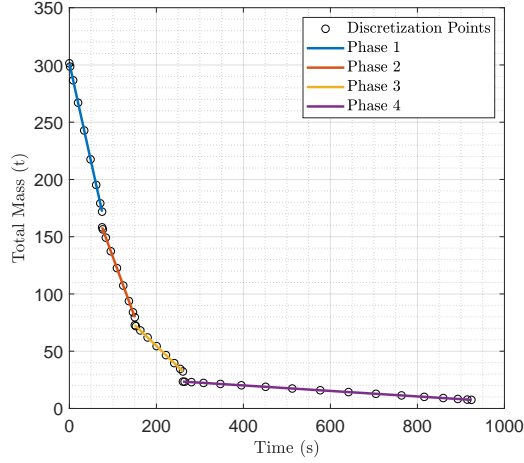


Fig. 7 Example IV.A- Mass profile vs time.

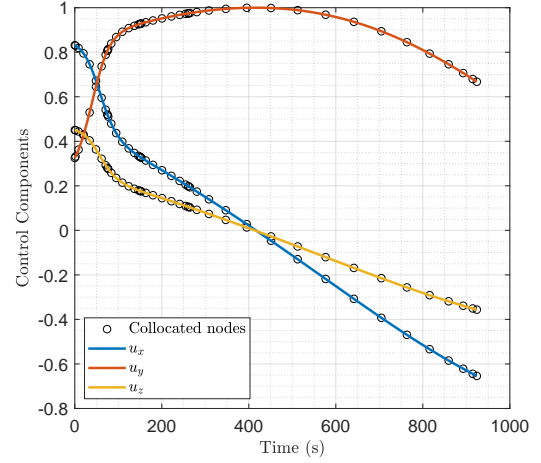


Fig. 8 Example IV.A- Cartesian components of control vs time.

be the optimal solution of state and control, but the same cannot be said about the dual variables. The solution of the dual variables, and, by extension, the Hamiltonian, are not unique to a given trajectory of state and control. For instance, in this particular case, the mass flow-rate is constant on every phase $\dot{m} = \text{Constant}$, and therefore the problem could be formulated by having open intermediate times, t_1 , t_2 and t_3 , and fixed endpoint masses (instead of fixed times and open masses), in which case the Hamiltonian would be zero everywhere. Nevertheless, the solution of state and control would be identical to the present one, as no additional degrees of freedom would be introduced.

By looking at Fig. 9, it is clear that the Hamiltonian yielded by SPARTAN does not concur with the solution of the original paper [11] in its entirety. The divergence of the Hamiltonian in phases 1 and 3, is thus, most likely, due to a slight difference in formulations between the two parties. As discussed above, although different formulations might affect the dual variable solution, they do not affect the solution of state and control, provided the degrees of freedom remain the same. In other words, the divergence of the Hamiltonian is not a relevant factor to determine optimality in this case. Ultimately, Fig. 9 shows that SPARTAN yields a phase-wise constant Hamiltonian and also that the Hamiltonian is zero during phase 4, implying that the solution is indeed optimal [10, 11].

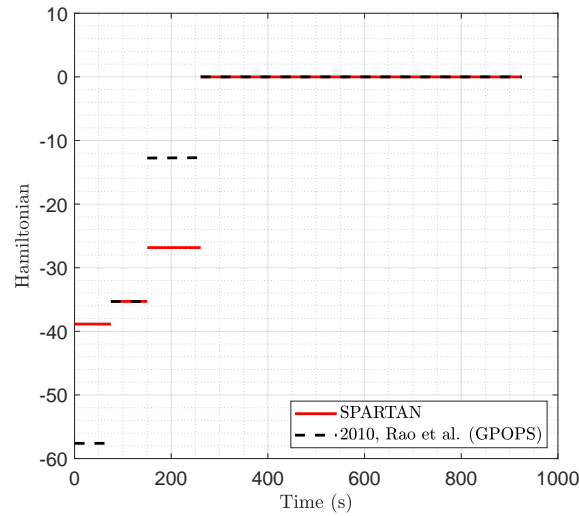


Fig. 9 Example IV.A- Hamiltonian comparison between SPARTAN and the source article [11].

To conclude the discussion, Table 3 presents the results of relevant parameters associated with the problem, namely

the final time of phase 4 and the final mass of the vehicle (Mayer cost). The results from the original article are also presented for reference. It can be noted that the final mass of the vehicle is only a couple hundred grams higher in the solution of SPARTAN with regards to the solution of the original article. This comparison reinforces our confidence in defining the trajectories virtually identical. Although the final time is unspecified in the original article, the results should be very similar as well due to the identical initial masses at the beginning of phase 4 and the constant mass flow rate during this phase. Ultimately, the results yielded by SPARTAN with the flipped Radau method are satisfactory.

Table 3 Comparison of relevant parameter results between SPARTAN and 2010, Rao et al. [11].

Parameter	SPARTAN	2010, Rao et al.	Unit
Total time of flight, t_f	924.13043	unspecified	s
Final mass, $m(t_f)$	7529.9284	7529.7123	kg

B. Apollo 11's Saturn V Ascent

The second example tries to consider a more realistic scenario with respect to the previous one. The recent 50th anniversary of the historical Moon landing provided the perfect opportunity to have a look at the Apollo 11 mission, and to try to reconstruct its mission through the proposed algorithm. As it is well-known such a magnificent achievement could have not been successfully completed without the Saturn V rocket [35]. The rocket consists of 3 stages: the S-IC stage, having 5 F-1 non-throttleable engines, each of them generating 6700 kN for a total of more than 33 MN, the S-II stage, with 5 J-2 engines, each generating 1 MN, and the S-IVB stage, equipped with another J-2 engine. For a more detailed reconstruction of the Apollo 11 mission the reader is suggested to refer to Ref [35].

Here we will focus on the ascent part, that is, from launch to the Earth orbit injection. By looking at the original NASA documentation [35] it was decided to model the problem with five different phases.

- Phase 1 In the first phase the 5 F-1 engines provide a constant thrust for about 135 s, when an altitude of 44 km is reached.
- Phase 2 At this point the central engine is shut down, and the ascent continues with the 4 lateral engines.
- Phase 3 When the first-stage separation occurs, the rocket continues to increase its speed sustained by the thrust of the 5 S-II engines and reaches an altitude of 180 km with a speed of 5.3 km/s.
- Phase 4 After 460 s a MECO command for the central J-2 engine is issued, and as in the previous phase, the lateral 4 engines continue to provide thrust until a time of about 550 s.
- Phase 5 Finally, a second stage separation event takes place, and the single J-2 engine generates thrust until orbital conditions are achieved, at about 710 s.

The equations of motion used to describe the problem are formulated with respect to the Earth-Centered, Earth-Fixed reference frame. Although more choices are available (e.g., inertial-based representations), this formulation was adopted to make the comparison with the original documentation easier, which contains the trajectory in tabular data in this reference frame. Note that for the Saturn-V the engines were non-throttleable. Therefore the fuel consumption cannot be optimized and the cost function adopted was aiming at the minimization of the control rate effort.

$$\mathcal{J} = \int_{t_0}^{t_f} \left(\dot{\theta}(t)^2 + \dot{\psi}(t)^2 \right) dt \quad (48)$$

The equations of motion, expressed in ECEF coordinates, are the following

$$\begin{cases} \dot{\mathbf{r}} &= \mathbf{v} \\ \dot{\mathbf{v}} &= -\frac{\mu}{\|\mathbf{r}\|^3} \mathbf{r} + \frac{T}{m} \mathbf{u} + \frac{\mathbf{D}}{m} - 2\boldsymbol{\Omega}_{\oplus} \times \mathbf{v} - \boldsymbol{\Omega}_{\oplus} \times \boldsymbol{\Omega}_{\oplus} \times \mathbf{r} \\ \dot{m} &= -\frac{T}{g_0 I_{sp}} \end{cases}$$

where the control \mathbf{u} is the Cartesian unit vector coming from the proper transformation of roll, pitch and yaw angles,

with roll assumed constant. The initial and final conditions are defined as

$$\begin{aligned}\mathbf{r}(t_0^{(1)}) &= [916\,245.39 \quad -5\,537\,048.67 \quad 3\,020\,181.60]^T \text{m} \\ \mathbf{v}(t_0^{(1)}) &= [0 \quad 0 \quad 0]^T \text{m/s} \\ m(t_0^{(1)}) &= 2\,902\,328 \text{ kg}\end{aligned}$$

and

$$\begin{aligned}\mathbf{r}(t_F^{(5)}) &= [3\,360\,940.44 \quad -4\,410\,922.63 \quad 3\,510\,251.57]^T \text{m} \\ \mathbf{v}(t_F^{(5)}) &= [5825.78 \quad 4543.92 \quad 133.38]^T \text{m/s} \\ m(t_F^{(5)}) &= 166\,450 \text{ kg}\end{aligned}$$

with the latter coming from the constraints required to achieve Earth orbital injection at $t_F = 709.3$ s. Note that quantities like the thrust, the massflow, and the specific impulse are modeled as look-up table based on the data plotted in [35], Figs. 5-3, 6-3, while the drag coefficient as function of the Mach number was obtained by looking at the reconstruction work performed by Braeunig [36]. Finally, for the atmosphere, although not available at the time of the Apollo 11 mission, a US-76 model has been adopted. [37].

Results are depicted in Figs. 10-13.

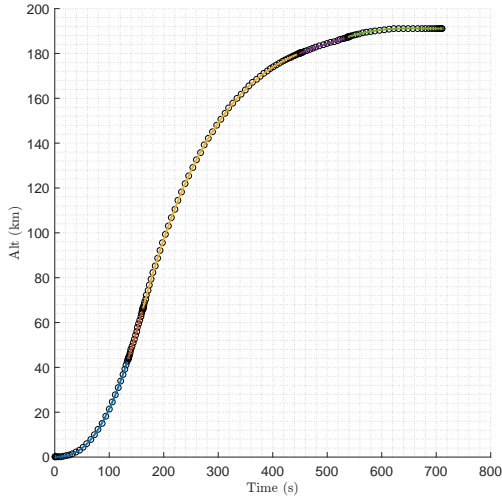


Fig. 10 Example IV.B- Altitude profile.

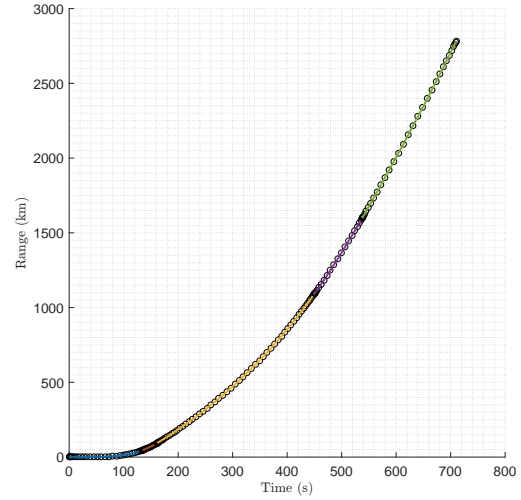


Fig. 11 Example IV.B- Range profile.

Figures 10 and 11 show the altitude and the range flown during the ascent. The black dashed lines identify the several phases of flight. The final altitude and range are fully consistent with the ones described in the original documentation, with the third stage correctly reaching orbital conditions at about 190 km of altitude after 710 s of flight. The corresponding controls are depicted in Fig. 12. Note that only pitch and yaw are used here as the engines were non-throttleable, and were therefore treated as a look-up table provided to the optimizer. Finally Fig. 13 shows the groundtrack of the ascent vehicle. The interface conditions representing the beginning of the parking orbit were successfully reached.

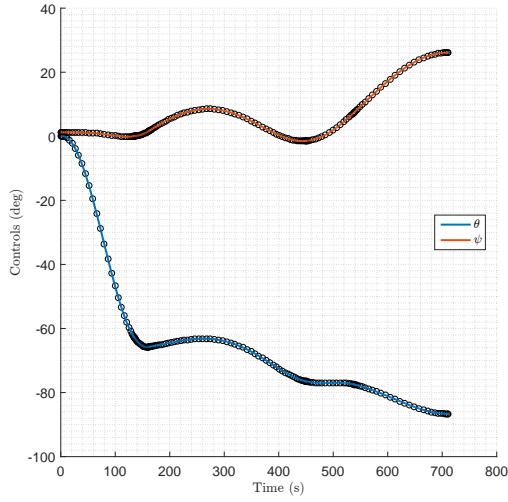


Fig. 12 Example IV.B- Pitch and yaw profiles.

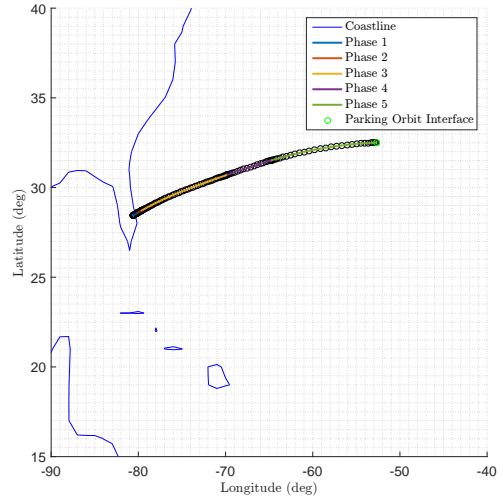


Fig. 13 Example IV.B- Groundtrack profile.

C. Reusable Rocket with In-flight re-ignition

The third example is a first application of descent-and-landing reusable technologies adapted from [38]. In this case we optimize the descent and landing phase of a reusable rocket with in-flight computation of the re-ignition time.

In this problem, the main stage of a reusable launch vehicle (RLV) is to be landed vertically at a target site. The booster is in free-fall at $t = t_0 = 0$, and the trajectory is composed of two distinct phases:

Phase 1 Coast phase. At $t = t_0$ the booster is in free-fall at an altitude of 15 km. The engines are shut off during all of the first phase.

Phase 2 Landing phase. The engines ignite at $t = t_1$ (unknown time) and the rocket is controllable. The booster lands and the engines are shut down at the unknown final time $t = t_f$.

The optimal control problem is stated as follows. Minimise the cost functional:

$$\mathcal{J} = -m(t_f^{(2)}) \quad (49)$$

subject to the equations of motion, expressed in NED:

$$\begin{cases} \dot{X} &= R_e \cdot V_N / r \\ \dot{Y} &= R_e \cdot V_E / (r \cos \phi) \\ \dot{H} &= -V_D \\ \dot{V}_N &= \dot{V} \cos \gamma \cos \psi - \dot{\gamma} V \sin \gamma \cos \psi - \dot{\psi} V \cos \gamma \sin \psi \\ \dot{V}_E &= \dot{V} \cos \gamma \sin \psi - \dot{\gamma} V \sin \gamma \sin \psi + \dot{\psi} V \cos \gamma \cos \psi \\ \dot{V}_D &= -\dot{V} \sin \gamma - \dot{\gamma} V \cos \gamma \\ \dot{m} &= -\|P\| / (I_{sp} g_0) \\ \dot{P} &= u \end{cases} \quad (50)$$

the event constraints

$$\begin{cases} X(t_0^{(1)}) = -2400 \text{ m} \\ Y(t_0^{(1)}) = -2800 \text{ m} \\ H(t_0^{(1)}) = 15000 \text{ m} \\ V(t_0^{(1)}) = 400 \text{ m/s} \\ \gamma(t_0^{(1)}) = -70 \text{ deg} \\ \psi(t_0^{(1)}) = 50 \text{ deg} \\ m(t_0^{(1)}) = 35000 \text{ kg} \end{cases} \quad \begin{cases} -0.5 \text{ m} \leq X(t_f^{(2)}) \leq 0.5 \text{ m} \\ -0.5 \text{ m} \leq Y(t_f^{(2)}) \leq 0.5 \text{ m} \\ 0 \leq H(t_f^{(2)}) \leq 0.2 \text{ m} \\ 0 < V(t_f^{(2)}) \leq 0.2 \text{ m/s} \\ -90 \text{ deg} < \gamma(t_f^{(2)}) \leq -89.5 \text{ deg} \\ \psi(t_f^{(2)}) = \text{free} \\ 25000 \text{ kg} \leq m(t_f^{(2)}) \end{cases} \quad (51)$$

And the path constraints

$$\mathbf{P}_{min} \leq \|\mathbf{P}\| \leq \mathbf{P}_{max} \quad (52)$$

$$|\dot{P}_i| \leq \dot{P}_{i,max} \quad i = x, y, z \quad (53)$$

with

$$\mathbf{P}_{min} = \begin{cases} 0, & t \in [t_0, t_1] \\ 400 \text{ kN}, & t \in [t_1, t_f] \end{cases} \quad \mathbf{P}_{max} = \begin{cases} 0, & t \in [t_0, t_1] \\ 800 \text{ kN}, & t \in [t_1, t_f] \end{cases}$$

$$\dot{P}_{i,max} = \begin{cases} 1000 \text{ N/s}, & i = x \\ 2000 \text{ N/s}, & i = y \\ 667.6 \text{ N/s}, & i = z \end{cases} \quad (54)$$

$$\begin{cases} \dot{V} &= -P_x/m - D/m - \mu \sin \gamma / r^2 \\ \dot{\gamma} &= P_y/(mV) + V \cos \gamma / r - \mu \cos \gamma / (Vr^2) + 2\Omega_{\oplus} \sin \psi \cos \phi \\ \dot{\psi} &= P_z/(mV \cos \gamma) + V \tan \phi \cos \gamma \sin \psi / r - 2\Omega_{\oplus} (\cos \psi \tanh \gamma \cos \phi - \sin \phi) \end{cases}$$

and

$$\begin{cases} V_N &= V \cos \gamma \cos \psi \\ V_E &= V \cos \gamma \sin \psi \\ V_D &= -V \sin \gamma \\ \|\mathbf{P}\| &= \sqrt{P_x^2 + P_y^2 + P_z^2} \\ D &= \frac{1}{2} \rho_0 e^{-kH} V^2 S_{ref} C_D \end{cases} \quad \begin{cases} V &= \sqrt{V_N^2 + V_E^2 + V_D^2} \\ \gamma &= -\arctan \left(V_D / \sqrt{V_N^2 + V_E^2} \right) \\ \psi &= \arctan (V_E / V_N) \\ \phi &= X / R_e \\ r &= H + R_e \end{cases}$$

Where X is the crossrange (displacement along meridians); Y is the downrange (displacement along parallels); H is the altitude of the spacecraft; V_N , V_E and V_D are the North-East-Down components of the velocity vector, respectively; m is the mass of the vehicle; $\mathbf{P} = [P_x, P_y, P_z]^T$ is the thrust vector expressed relative to the velocity vector: P_x always pointing in the opposite direction of $\mathbf{V} = [V_N, V_E, V_D]^T$, P_y positive in the direction of increasing flight path angle, and P_z positive in the direction of increasing heading angle. Finally, the controls are set to be the thrust rates, as shown in the last expression of Eq. (50), $\mathbf{u} = [\dot{P}_x, \dot{P}_y, \dot{P}_z]^T$. Other relevant variables and quantities are: the norm of the velocity vector, $V = \|\mathbf{V}\|$; the flight path angle, γ ; the heading angle, ψ ; latitude and longitude, ϕ and θ respectively.

The Cartesian formulations of the path constraints of thrust and thrust-rate expressed in Eqs. (52) and (53) respectively, are presented for the sake of simplicity and harmony with the original article, in reality both the thrust and the thrust-rates were modulated in spherical coordinates instead.

Finally, the linkage conditions are omitted because there are no expected discontinuities in the states of the system ($\Delta \mathbf{X}^{(1:2)} = \mathbf{0}$).

The constants used in this problem are expressed in Table 4. Note that some of the needed parameters were selected based on typical assumptions for this class of rockets and mission profile, whenever not available in the corresponding reference, (e.g., the thrust rate in Eq. (54) or some environmental parameters, described in Table 4). The problem was discretized with 6 nodes in phase 1, and 20 nodes in phase 2. The results are presented in Figs. 14 through 19.

Figures 14 and 15 present the altitude and the horizontal coordinates of the spacecraft, respectively. In Fig. 14 it can be seen a tangential approach to the time axis, indicating a soft landing; and in Fig. 15 both downrange and crossrange tend to the zero target.

Figures 16 and 17 represent the vertical and horizontal components of velocity, respectively. By inspection of Fig. 16 one can verify the increasing downward velocity during the coast phase followed by a decay of this component during the powered phase until zero is reached. In Fig. 17 the horizontal velocity components decay slightly during the coast phase as a result of the aerodynamic drag force acting on the vehicle and these components tend to zero at the end of the trajectory.

Table 4 Relevant constants and parameters for the reusable rocket with in-flight re-ignition problem.

Constant	Value	Unit
Specific impulse, I_{sp}	250	s
Reference surface area, S_{ref}	8	m ²
Air density at sea level, ρ_0	1.225	kg/m ³
Drag coefficient, C_D	0.25	1
Inverse, density scale height, k	$\frac{1}{7109}$	1/m
Earth radius, R_e	6378145	m
Earth rotation rate, Ω_{\oplus}	$7.29211585 \times 10^{-5}$	rad/s
Standard gravitational parameter, μ	3.986012×10^{14}	m ³ /s ²
Gravity acceleration at sea level, g_0	9.80665	m/s ²
Phase 1 initial time, t_0	0	s

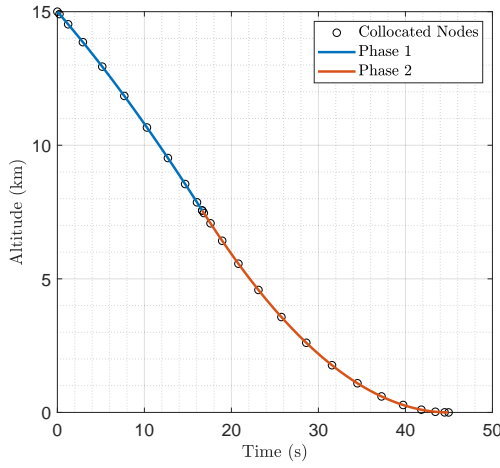


Fig. 14 Example IV.C- Altitude with time.

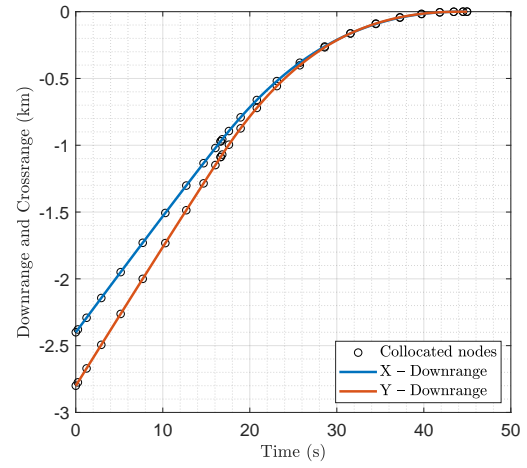


Fig. 15 Example IV.C- Downrange and cross-range vs time.

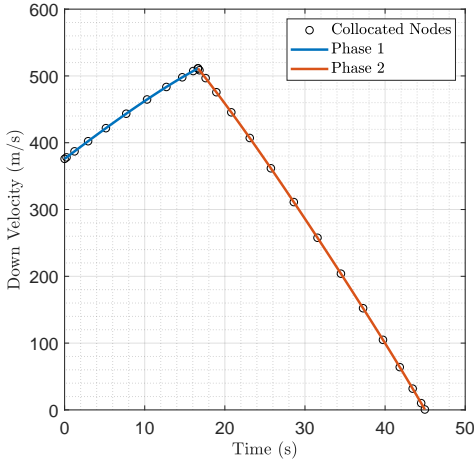


Fig. 16 Example IV.C- Vertical velocity (descending) with time.

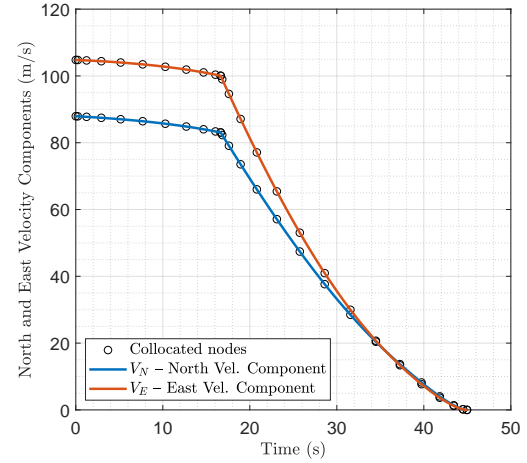


Fig. 17 Example IV.C- North and East velocity components with time.

Finally, the flight path concurrent component of thrust is illustrated in Fig. 18 and the thrust components transverse

to the flight path are shown in figure 19. The most prominent feature of the plots is, of course, the discontinuity as a result of the sudden engine ignition at $t = t_1$. Because this problem has a "bang-bang" solution on the thrust rates, the increase-decrease corner in P_y near the 40 s mark will "sharpen" by increasing the number of nodes used in the second phase. One notable instance of evident disagreement when comparing the solution with the source article is the profile of the thrust component P_z .

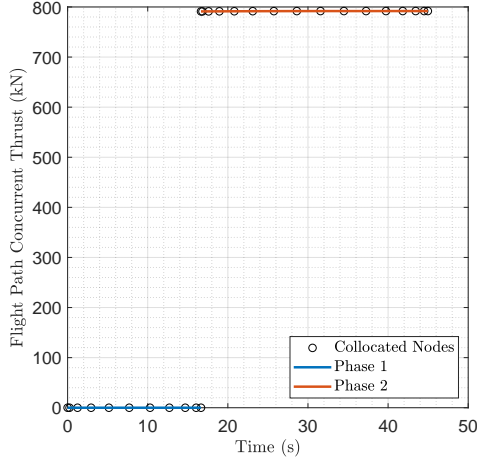


Fig. 18 Example IV.C- Flight path concurrent thrust component with time.

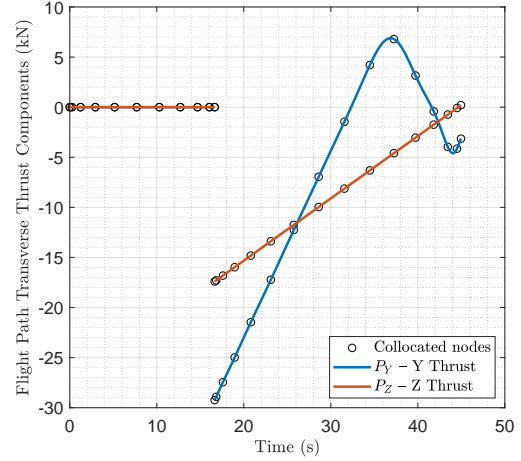


Fig. 19 Example IV.C- Flight path transverse thrust components with time.

Key output values of the solution are the final mass of the vehicle, 25.98×10^3 kg; the ignition time, $t_1 = 17.6038$ s (transition time from phase 1 to phase 2); and the final time of the trajectory, $t_f = 43.8545$ s. The results presented in Figs. 14 through 19 show satisfactory agreement with [38]. Slight discrepancies in the results may be due to differences in the choice of some constants, whose values were part of the assumptions made for this work.

D. Falcon 9 First Stage Recovery

The last example to consider is an adaptation from [39]. This example is concerned with the recovery of the first stage of an orbital launcher via vertical landing, where the landing target is located close to the launching site (return-to-launch-site scenario). The vehicle in question is based on the characteristics of the main booster of SpaceX's Falcon 9 rocket.

At the beginning of the trajectory stage separation is assumed to have already occurred and the booster is assumed to have performed the "turn back" attitude manoeuvre such that the boost-back-burn may begin. The trajectory is composed of three phases:

- Phase 1 Boost-back. At $t = t_0$ the rocket is in mid air with ascending linear momentum. Three engines (out of nine) ignite and the thrust direction is constrained to point strictly horizontally. In this phase the rocket reverts the horizontal direction of flight. The duration of the boost-back burn, $t = t_1$, is a known, fixed parameter.
- Phase 2 Coast arc. At $t = t_1$ the engines shut down and remain off during all of phase 2. The vehicle takes a parabolic flight profile during this phase. It is taken for granted that an attitude justification takes place during this phase in order to align the thrust vector with the velocity vector at the beginning of phase 3.
- Phase 3 Landing phase. At $t = t_2$ one engine (out of nine) ignites. The vehicle is now controllable and the booster makes its way to the target site. Touch-down happens at $t = t_f$. Both t_2 and t_f are unknown variables to be determined.

The equations of motion are taken from the original article [39], and are formulated in the target-centered reference frame. Throughout the trajectory, both position and velocity are modelled in Cartesian coordinates with the displacement reference frame centred at the landing target location. Downrange is represented by the horizontal axis, X , increasing from left to right, and altitude is represented by the vertical axis, Y , increasing from bottom to top. The Cartesian components of velocity are positive in the direction of increasing position coordinates, respectively. The Earth is assumed to be flat and non rotating, and the acceleration of gravity is assumed to be constant throughout the trajectory (invariant with altitude).

The optimal control problem can be formulated as follows. Minimize the cost functional:

$$\mathcal{J} = -m(t_f) \quad (55)$$

associated with the three phased dynamic system

$$\dot{x} = v_x \quad (56)$$

$$\dot{y} = v_y \quad (57)$$

$$\dot{v}_x = \frac{T}{m} k \cos \theta - \frac{D}{m} \frac{v_x}{v} \quad (58)$$

$$\dot{v}_y = \frac{T}{m} k \sin \theta - \frac{D}{m} \frac{v_y}{v} - g_0 \quad (59)$$

$$\dot{m} = -\frac{T}{I_{sp} g_0} k \quad (60)$$

and subject to the event constraints

$$x(t_0) = 36.022 \text{ km} \quad (61) \quad x(t_f) = 0 \text{ km} \quad (66)$$

$$y(t_0) = 60.708 \text{ km} \quad (62) \quad y(t_f) = 0 \text{ km} \quad (67)$$

$$v_x(t_0) = 1.052 \text{ km/s} \quad (63) \quad v_x(t_f) = 0 \text{ km/s} \quad (68)$$

$$v_y(t_0) = 1.060 \text{ km/s} \quad (64) \quad -0.5 \text{ m/s} \leq v_y(t_f) \leq 0.5 \text{ m/s} \quad (69)$$

$$m(t_0) = 76501 \text{ kg} \quad (65) \quad m(t_f) = \text{free} \quad (70)$$

with

$$D = \frac{1}{2} \rho_0 e^{-\frac{y}{h_0}} C_D \pi \frac{d^2}{4} v^2 \quad (71)$$

$$v^2 = v_x^2 + v_y^2 \quad (72)$$

and

$$T = \begin{cases} \frac{1}{3} T_{LO} & t \in [t_0, t_1] \\ 0 & t \in [t_1, t_2] \\ \frac{1}{9} T_{LO} & t \in [t_2, t_3] \end{cases} \quad (73)$$

where x is downrange, y is altitude, v_x is horizontal velocity, positive in the direction of increasing downrange, v_y is vertical velocity, positive in the direction of increasing altitude, m is the mass of the vehicle, k is the engine throttle, θ is the tilt angle representing the direction of thrust ($\theta = 0$ deg points in the direction of increasing v_x , and $\theta = 90$ deg points in the direction of increasing v_y), D is the aerodynamic drag force, γ is the flight path angle, ρ_0 is the air density at sea level, h_0 is the density scale height, C_D is the drag coefficient, d is the diameter of the vehicle and v is the norm of the velocity vector. The linkage conditions are omitted because there are no expected discontinuities on the states of the systems in any phase transition. Table 5 shows the constants and parameters used in this problem.

During phase 1 the trajectory is completely determined because the control sequence is known and the endpoint times are fixed, and in phase 2 the vehicle is uncontrollable due to the throttle being constrained to zero. Ultimately, the only degrees of freedom available to optimize the trajectory, therefore, are the engine reignition time, t_2 , the total time of flight, t_f and the direction of the thrust vector during phase 3.

With this simplified formulation one makes sure that the problem is feasible, therefore an optimal solution is guaranteed to exist and it is possible to focus on the ability of the algorithm to find it. A simplified formulation of the problem is adequate in this case because there is no interest in studying the feasibility of the problem, there is only interest in validating the flipped Radau method.

This problem was solved using 10, 8 and 12 collocation nodes in phases 1, 2 and 3, respectively. The NLP solver used was IPOPT [18], and the method used to compute partial derivative was the complex step differentiation method [21, 34]. The results are depicted in Figs. 20 through 25. In addition the resulting parameters of unknown times and final mass are presented in table 6 including the references of the original article.

Figures 20 and 21 illustrate the Cartesian components of position, and the Cartesian components of velocity with time, respectively. The phase transition points are visible through the increase in density of collocation nodes. With

Table 5 Relevant constants and parameters for the Falcon 9 first stage recovery problem.

Constant	Value	Unit
Specific impulse, I_{sp}	282	s
Lift-off Thrust, T_{LO}	5886	kN
Rocket diameter, d	3.66	m
Drag Coefficient, C_D	0.75	1
Density scale height, h_0	7500	m
Gravity acceleration at seal level, g_0	9.80665	m/s ²
Air density at sea level, ρ_0	1.225	kg/m ³
Phase 1 initial time, t_0	0	s
Phase 1 / Phase 2 transition time, t_1	40.8	s
Dry mass, m_{dry}	25600	kg

regards to Fig. 21, the phase transitions are clearly pronounced due to the removable discontinuities present in each component. The horizontal component of velocity decreases linearly during the first phase and goes from positive to negative, inverting the direction of flight, which is indicative of the boost-back burn phase. This component stays constant during the coasting phase, as expected. It is noticeable that during phases 1 and 2, the vertical component of velocity decreases linearly, without noticeable discontinuities, indicating that this component has been subject to a constant acceleration during these two phases, undoubtedly the acceleration of gravity. It is also visible that the vertical component of velocity goes from positive to negative close to the 110 s mark, inverting the direction flight. Both velocity components go to zero during the landing phase, as expected. In contrast, regarding Fig. 20, the position does not contain removable discontinuities, this is due to the fact that position is the integration of velocity through time, resulting in a curve that is "one degree" smother. One relevant remark is that both the downrange and the altitude are tangent to the time axis at the endpoint of the trajectory, indicating a smooth landing.

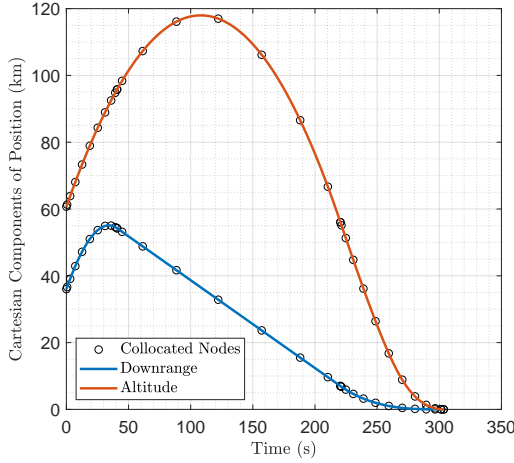


Fig. 20 Example IV.D- Downrange and altitude vs time.

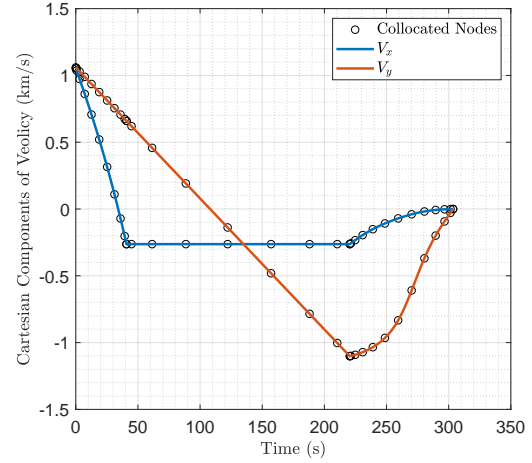


Fig. 21 Example IV.D- Cartesian components of velocity vs time.

The angle indicating the thrust direction (control) is presented in Fig. 22. It can be seen that the direction of thrust is constrained to 180 deg during phase 1, and constrained to 0 deg during phase 2 (during phase 2 the thrust direction is inconsequential due to the throttle being constrained to zero). Finally, in the landing phase the direction of thrust is allowed to vary, and the angle draws a curve that approaches 90 deg, indicating that the thrust points vertically at the end of the trajectory.

The trajectory profile of the vehicle is shown in Fig. 23, where the downrange and the altitude are plotted against each other. The boost-back phase is shown in blue, the coasting arc is shown in orange and the landing phase is represented in yellow. Curiously, the vehicle traces a path that vaguely resembles a shepherd's staff. The plot clearly shows that at the beginning of the trajectory the vehicle is travelling from left to right and with ascending velocity. The

boost-back burn inverts the horizontal motion of the vehicle and during the coasting arc, the vehicle takes a parabolic flight due to being in complete free-fall. Finally, in the landing phase the vehicle approaches the zenith of the target and the flight path angle gets closer and closer to being vertical. Intuitively speaking, the trajectory follows a predictable path.

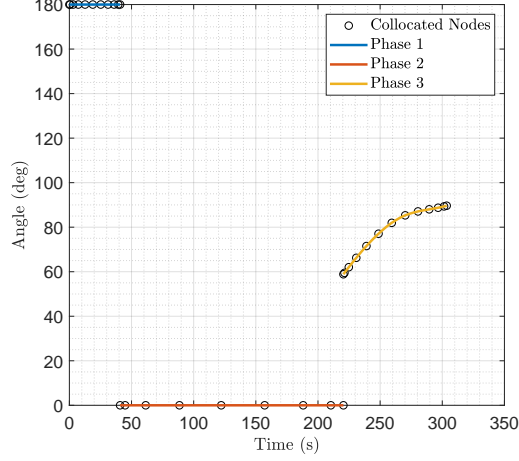


Fig. 22 Example IV.D- Thrust direction with time.

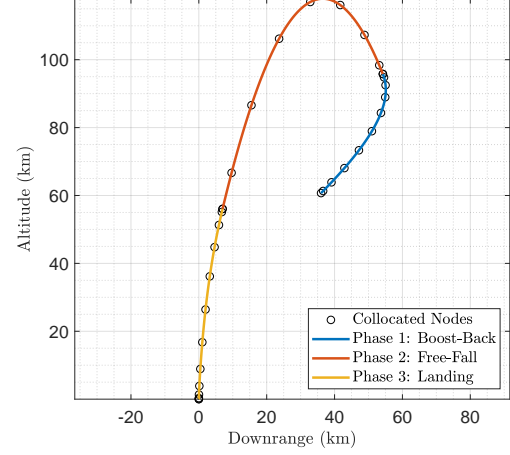


Fig. 23 Example IV.D- Altitude vs downrange.

Fig. 24 presents a comparison between the trajectory yielded by SPARTAN and the solution from the original article. It can be seen that the trajectories are quite similar, having an identical maximum altitude. The solutions diverge slightly during the descent of the vehicle, and the trajectory from SPARTAN approaches the target at a steeper flight path angle. This figure is presented in order to confirm the plausibility of the scenario, given that the two problem formulations are similar. However, the two trajectories must be judged independently for their optimality, as the corresponding problem formulations are in fact not the same, having a different number of degrees of freedom and a different cost functional.

Finally, Fig. 25 shows the Hamiltonian associated with the trajectory. By inspection of the figure, one can confidently assert that the Hamiltonian is phase-wise constant. In addition, one can also verify that the Hamiltonian is zero in phases 2 and 3. This result is satisfying because it goes in agreement with what is expected of an optimal solution when the Hamiltonian is not an explicit function of time [10, 11, 40]. Overall, the Hamiltonian brings further confirmation that the solution is optimal.

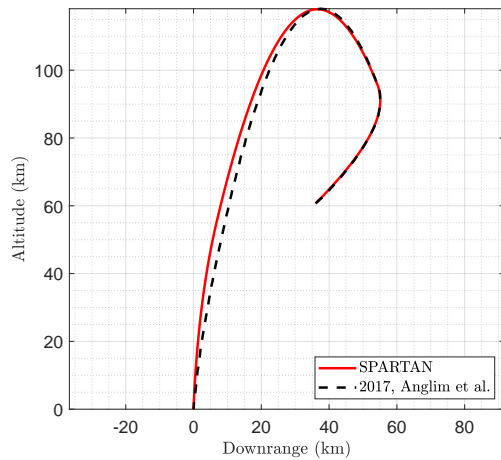


Fig. 24 Example IV.D- 2-D trajectory comparison.

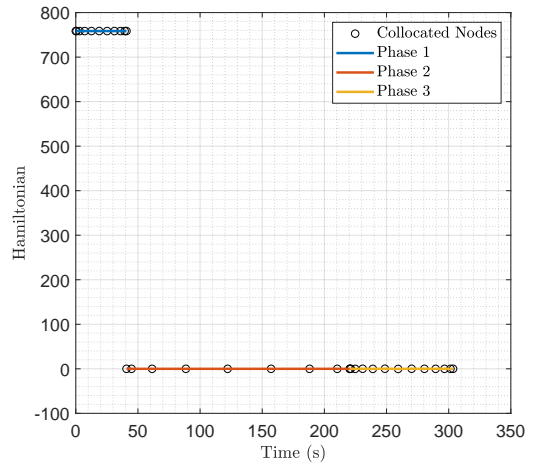


Fig. 25 Example IV.D- Hamiltonian vs time.

To conclude the discussion, table 6 presents the values of pertinent optimization variables obtained with SPARTAN. The results of the original article [39] are also presented for comparison. Briefly speaking, and disregarding the duration of the boost-back phase, the trajectory yielded by SPARTAN consists of a longer duration of coasting arc, and a shorter duration of powered descent when compared to the solution of the original article. This also results in a significantly higher final mass of the vehicle. The results are satisfactory overall.

Table 6 Comparison of relevant parameter results between SPARTAN and Anglim et al. [39].

Parameter	SPARTAN	2017, Anglim et al.	Unit
Boost-back burn duration, t_1	40.8	40.5684	s
Landing burn start time, t_2	220.4583	216.9112	s
Total time of flight, t_f	303.5469	307.8189	s
Final mass, $m(t_f)$	27905.5554	26221.1488	kg

V. Conclusions

In this paper we proposed a unified framework based on multi-phase optimization to deal with rocket ascent- and descent-trajectory scenarios. The approach was conceived to deal with systematic difficulties arising in modeling classical rocket missions with standard unique-phase optimal-control packages. Examples of such cases are stage separations and therefore discontinuities in some of the states (like the mass), or flight profiles where different types of motion are experienced. A typical example is represented by a coasting arc after the commanded shutdown of the engine followed by a powered phase to perform pinpoint landing. To validate the methodology we analyzed problems of different complexity ranging from the classical ascent of Saturn-V to the return of a reusable stage inspired by Falcon-9 rocket.

The benefits of transcribing the problem by using the flipped Radau method are twofold. On one hand all the valuable properties of pseudospectral methods, like the easiness of implementation and the spectral convergence, are exploited for the development of the multiphase algorithm. On the other hand we exploit the heritage of SPARTAN, the tool developed by the German Aerospace Center, which was in its previous implementation able to cope with one-phase problems only, and which represented the ideal ground for further developing the methodology.

Results confirm that the proposed approach is a valid methodology for rapidly prototyping solutions covering a wide range of scenarios with little effort, and therefore becomes a viable tool to design and analyze missions involving different sets of equations of motion as well as state discontinuities. Future work will include the introduction of isoperimetric constraints and non-causal phases to be able to model multiple bodies at the same time.

References

- [1] Benson, D. A., Huntington, G. T., Thorvaldsen, T. P., and Rao, A. V., "Direct Trajectory Optimization and Costate Estimation via an Orthogonal Collocation Method," *Engineering Notes, Journal of Guidance, Control, and Dynamics*, Vol. 29, No. 6, 2006, pp. 1435–1440. doi: [10.2514/1.20478](https://doi.org/10.2514/1.20478). [I](#)
- [2] Elnagar, G., Kazemi, M. A., and Razzaghi, M., "The pseudospectral Legendre method for discretizing optimal control problems," *IEEE Transactions on Automatic Control*, Vol. 40, No. 10, 1995, pp. 1793–1796. doi: [10.1109/9.467672](https://doi.org/10.1109/9.467672). [I](#)
- [3] Fahroo, F., and Ross, I. M., "Costate Estimation by a Legendre Pseudospectral Method," *Journal of Guidance, Control, and Dynamics*, Vol. 24, No. 2, 2001, pp. 270–277. doi: [10.2514/2.4709](https://doi.org/10.2514/2.4709). [I](#)
- [4] Fahroo, F., and Ross, I. M., "Direct Trajectory Optimization by a Chebyshev Pseudospectral Method," *Journal of Guidance, Control, and Dynamics*, Vol. 25, No. 1, 2002, pp. 160–166. doi: [10.2514/2.4862](https://doi.org/10.2514/2.4862).
- [5] Ross, I. M., and Fahroo, F., "Legendre Pseudospectral Approximations of Optimal Control Problems," *Springer*, Vol. 295, 2003, pp. 327–342. doi: [10.1007/978-3-540-45056-6_21](https://doi.org/10.1007/978-3-540-45056-6_21). [I](#), [III.C](#)
- [6] Gong, Q., Kang, W., Bedrossian, N. S., Fahroo, F., Sekhavat, P., and Bollino, K., "Pseudospectral Optimal Control for Military and Industrial Applications," *2007 46th IEEE Conference on Decision and Control*, IEEE, 2007, pp. 4128–4142. doi: [10.1109/cdc.2007.4435052](https://doi.org/10.1109/cdc.2007.4435052). [I](#)

- [7] Benson, D. A., “A Gauss Pseudospectral Transcription for Optimal Control,” Ph.D. thesis, Massachusetts Institute of Technology, 2004. [I](#)
- [8] Ross, I. M., Sekhavat, P., Fleming, A., Gong, Q., and Kang, W., “Pseudospectral Feedback Control: Foundations, Examples and Experimental Results,” *AIAA Guidance, Navigation, and Control Conference and Exhibit*, American Institute of Aeronautics and Astronautics, 2006, pp. 1–23. doi: [10.2514/6.2006-6354](#).
- [9] Fariba Fahroo and I. Michael Ross, “Advances in Pseudospectral Methods for Optimal Control,” *AIAA Guidance, Navigation, and Control Conference and Exhibit, Honolulu, USA, 2008*, 2008, pp. 1–23. doi: [10.2514/6.2008-7309](#). [I](#), [III.A](#), [III.B](#), [III.B](#)
- [10] Garg, D., “Advances in Global Pseudospectral Methods for Optimal Control,” Ph.D. thesis, University of Florida, 2011. [I](#), [IV.A](#), [IV.D](#)
- [11] Rao, A. V., Benson, D. A., Darby, C. L., Patterson, M. A., Francolin, C., Sanders, I., and Huntington, G. T., “Algorithm 902: GPOPS, A MATLAB Software for Solving Multiple-Phase Optimal Control Problems Using the Gauss Pseudospectral Method,” *ACM Transactions on Mathematical Software*, Vol. 37, No. 2, 2010, pp. 1–39. doi: [10.1145/1731022.1731032](#). [I](#), [III.C](#), [IV.A](#), [2](#), [IV.A](#), [5](#), [6](#), [IV.A](#), [9](#), [3](#), [IV.D](#)
- [12] Patterson, M. A., and Rao, A. V., “GPOPS-II: A MATLAB Software for Solving Multiple-Phase Optimal Control Problems Using hp-Adaptive Gaussian Quadrature Collocation Methods and Sparse Nonlinear Programming,” *ACM Transactions on Mathematical Software*, Vol. 41, No. 1, 2014, pp. 1–37. doi: [10.1145/2558904](#). [I](#), [IV.A](#), [1](#)
- [13] Becerra, V. M., “Solving complex optimal control problems at no cost with PSOPT,” *2010 IEEE International Symposium on Computer-Aided Control System Design*, IEEE, 2010, pp. 1391–1396. doi: [10.1109/cacsd.2010.5612676](#). [I](#)
- [14] Marco Sagliano and Stephan Theil and Vincenzo D’Onofrio and Michiel Bergsma, “SPARTAN: A Novel Pseudospectral Algorithm for Entry, Descent, and Landing Analysis,” *Advances in Aerospace Guidance, Navigation and Control*, Springer International Publishing, 2017, pp. 669–688. doi: [10.1007/978-3-319-65283-2_36](#). [I](#), [III.C](#)
- [15] N. S. Bedrossian and S. Bhatt and W. Kang and I. M. Ross, “Zero-propellant maneuver guidance,” *IEEE Control Systems Magazine*, Vol. 29, No. 5, 2009, pp. 53–73. doi: [10.1109/MCS.2009.934089](#). [I](#)
- [16] Gill, P. E., Wong, E., Murray, W., and Saunders, M. A., “User’s Guide for SNOPT Version 7.6: Software for Large-Scale Nonlinear Programming,” *University of California, San Diego*, 2017. [I](#)
- [17] Wächter, A., and Biegler, L., “On the Implementation of an Interior-Point Filter Line-Search Algorithm for Large-Scale Nonlinear Programming,” *Mathematical programming*, Vol. 106, 2005, pp. 25–57. doi: [10.1007/s10107-004-0559-y](#). [I](#)
- [18] Kawajir, Y., Laird, C., Vigerske, S., and Wächter, A., “Introduction to IPOPT: A tutorial for downloading, installing, and using IPOPT,” *Chicago Northwestern University*, 2015. [I](#), [IV.A](#), [IV.D](#)
- [19] Ross, I. M., “A Historical Introduction to the Covector Mapping Principle,” *AAS / AIAA Astrodynamics Specialist Conference, Tahoe, NV, USA, 2005*, pp. 1–21. [I](#)
- [20] Gong, Q., Ross, I. M., Kang, W., and Fahroo, F., “Connections Between The Covector Mapping Theorem and Convergence of Pseudospectral Methods for Optimal Control,” *Computational Optimization and Applications*, Vol. 41, No. 3, 2008, pp. 307–335. doi: [10.1007/s10589-007-9102-4](#). [I](#)
- [21] D’Onofrio, V., “Implementation of Advanced Differentiation Methods for Optimal Trajectory Computation,” Master’s thesis, Università Degli Studi Di Napoli “Federico II”, 2014. [I](#), [IV.A](#), [IV.D](#)
- [22] D’Onofrio, Vincenzo and Sagliano, Marco and Arslantaş, Yunus E., “Exact Hybrid Jacobian Computation for Optimal Trajectory Generation via Dual Number Theory,” *American Institute of Aeronautics and Astronautics*, 2015.
- [23] Sagliano, M., Theil, S., Bergsma, M., D’Onofrio, V., Whittle, L., and Viavattene, G., “On the Radau Pseudospectral Method: theoretical and implementation advances,” *CEAS Space Journal*, Vol. 9, No. 3, 2017, pp. 313–331. doi: [10.1007/s12567-017-0165-5](#). [I](#), [III.B](#), [III.B](#), [III.B](#)
- [24] Agamawi, Y. M., Hager, W. W., and Rao, A. V., “Mesh Refinement Method for Solving Bang-Bang Optimal Control Problems Using Direct Collocation,” *AIAA Scitech 2020 Forum*, 2020. doi: [10.2514/6.2020-0378](#). [I](#)
- [25] Rao, A. V., “A Survey of Numerical Methods for Optimal Control,” *AAS/AIAA Astrodynamics Specialist Conference, AAS Paper 09-334, Pittsburgh, PA, August 10 - 13, 2009*, pp. 1–32. [I](#)

- [26] Sagliano, M., “Performance analysis of linear and nonlinear techniques for automatic scaling of discretized control problems,” *Operations Research Letters*, Vol.42 Issue 3, May 2014, pp. 213-216, 2014. doi: [10.1016/j.orl.2014.03.003](https://doi.org/10.1016/j.orl.2014.03.003). **I**
- [27] Darby, C. L., and Rao, A. V., “A Mesh Refinement Algorithm for Solving Optimal Control Problems Using Pseudospectral Methods,” *American Institute of Aeronautics and Astronautics*, 2009. **I**
- [28] Sagliano, M., “Generalized hp Pseudospectral-Convex Programming for Powered Descent and Landing,” *Journal of Guidance, Control, and Dynamics*, Vol. 42, No. 7, 2019, pp. 1562–1570. doi: [10.2514/1.g003731](https://doi.org/10.2514/1.g003731).
- [29] Koeppen, N., Ross, I. M., Wilcox, L. C., and Proulx, R. J., “Fast Mesh Refinement in Pseudospectral Optimal Control,” *Journal of Guidance, Control, and Dynamics*, Vol. 42, No. 4, 2019, pp. 711–722. doi: [10.2514/1.G003904](https://doi.org/10.2514/1.G003904).
- [30] Gong, Q., Fahroo, F., and Ross, I. M., “Spectral Algorithm for Pseudospectral Methods in Optimal Control,” *Journal of Guidance, Control, and Dynamics*, Vol. 31, No. 3, 2008, pp. 460–471. doi: [10.2514/1.32908](https://doi.org/10.2514/1.32908). **I**
- [31] Wiesel, W., *Spaceflight dynamics*, Aphelion Press, Beavercreek, Ohio, 2010. **II.C**
- [32] Berrut, J.-P., and Trefethen, L. N., “Barycentric Lagrange Interpolation,” *SIAM Review*, Vol. 46, No. 3, 2004, pp. 501–517. doi: [10.1137/s0036144502417715](https://doi.org/10.1137/s0036144502417715). **III.A**
- [33] Betts, J. T., *Practical Methods for Optimal Control and Estimation Using Nonlinear Programming*, 2nd ed., SIAM, Philadelphia, 2010. **IV.A**
- [34] Martins, J. R. R., Sturdza, P., and Alonso, J. J., “The Complex-Step Derivative Approximation,” *ACM Transactions on Mathematical Software*, Vol. 29, No. 3, September 2003, Pages 245-262, 2003. doi: [10.1145/838250.838251](https://doi.org/10.1145/838250.838251). **IV.A, IV.D**
- [35] Group, S. F. E. W., “Saturn V Launch Vehicle Flight Evaluation Report AS-506,” Tech. Rep. NASA-TM-X-62558, NASA, 1969. **IV.B, IV.B**
- [36] Braeunig, R. A., “Saturn V Launch Simulation,” , 2013. URL <https://web.archive.org/web/20170313142729/http://www.braeunig.us/apollo/saturnV.htm>, retrieval date 05-May-2020. **IV.B**
- [37] NASA, “U.S. Standard Atmosphere 1976,” Tech. Rep. Technical Memorandum X-74335, NASA, 1976. **IV.B**
- [38] Wang, J., and Cui, N., “A Pseudospectral-Convex Optimization Algorithm for Rocket Landing Guidance,” *2018 AIAA Guidance, Navigation, and Control Conference*, American Institute of Aeronautics and Astronautics, 2018, pp. 1–18. doi: [10.2514/6.2018-1871](https://doi.org/10.2514/6.2018-1871). **IV.C, IV.C**
- [39] Anglim, K. S. G., Zhang, Z., and Gao, Q., “Minimum-Fuel Optimal Trajectory For Reusable First-Stage Rocket Landing Using Particle Swarm Optimization,” *World Academy of Science, Engineering and Technology, International Journal of Mechanical, Aerospace, Industrial, Mechatronic and Manufacturing Engineering*, Vol. 11, 2017, pp. 920–929. doi: [10.5281/ZENODO.1130268](https://doi.org/10.5281/ZENODO.1130268). **IV.D, IV.D, 6**
- [40] Arthur E. Bryson Jr. and Yu-Chi Ho, *Applied Optimal Control. Optimization, Estimation, and Control*, Hemisphere Publishing Corporation, Washington New York, 1975. **IV.D**